Chapter 20

Network Layer: Internet Protocol

20-1 INTERNETWORKING

In this section, we discuss internetworking, connecting networks together to make an internetwork or an internet.

Topics discussed in this section:

Need for Network Layer
Internet as a Datagram Network
Internet as a Connectionless Network

Figure 20.1 *Links between two hosts*

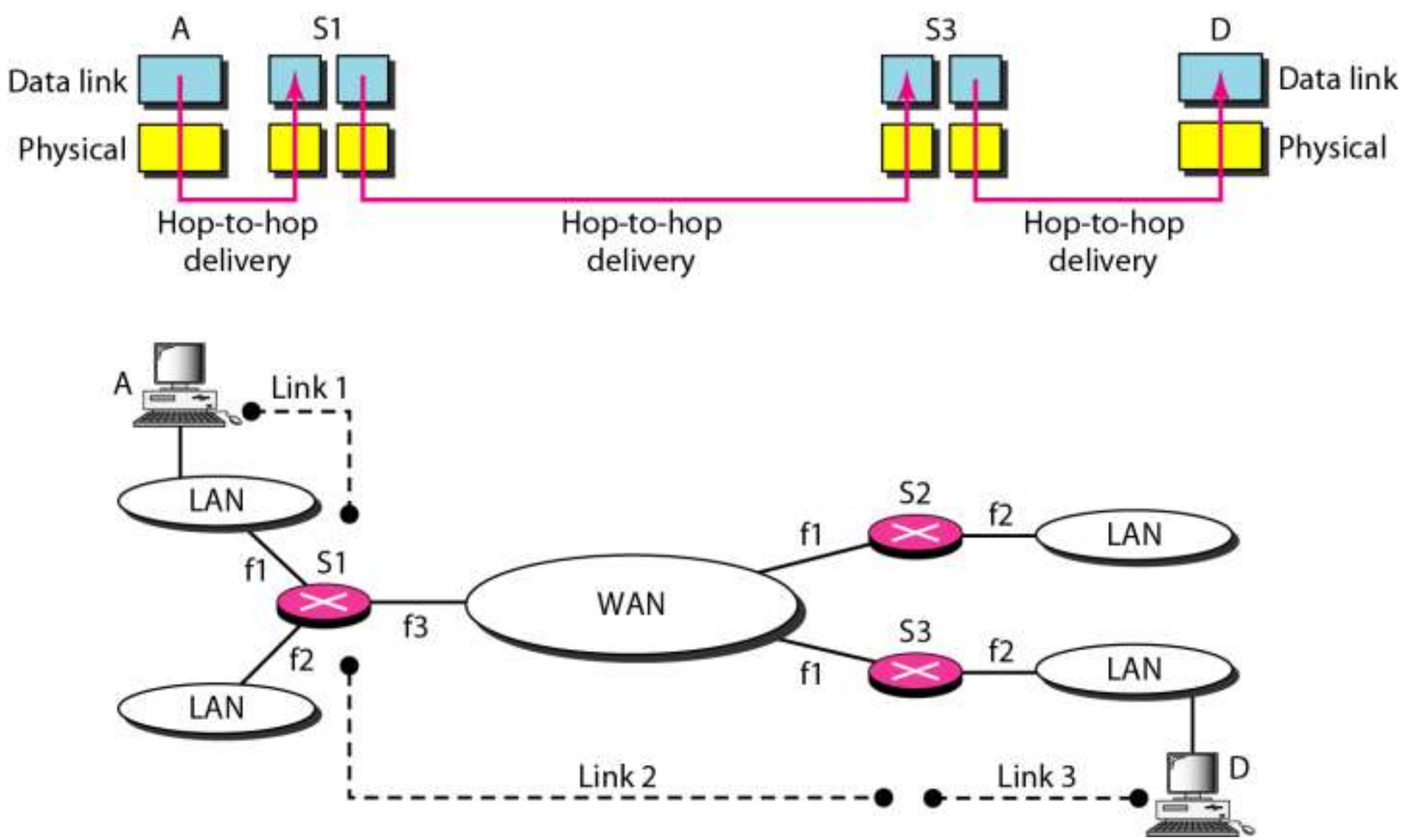


Figure 20.2 *Network layer in an internetwork*

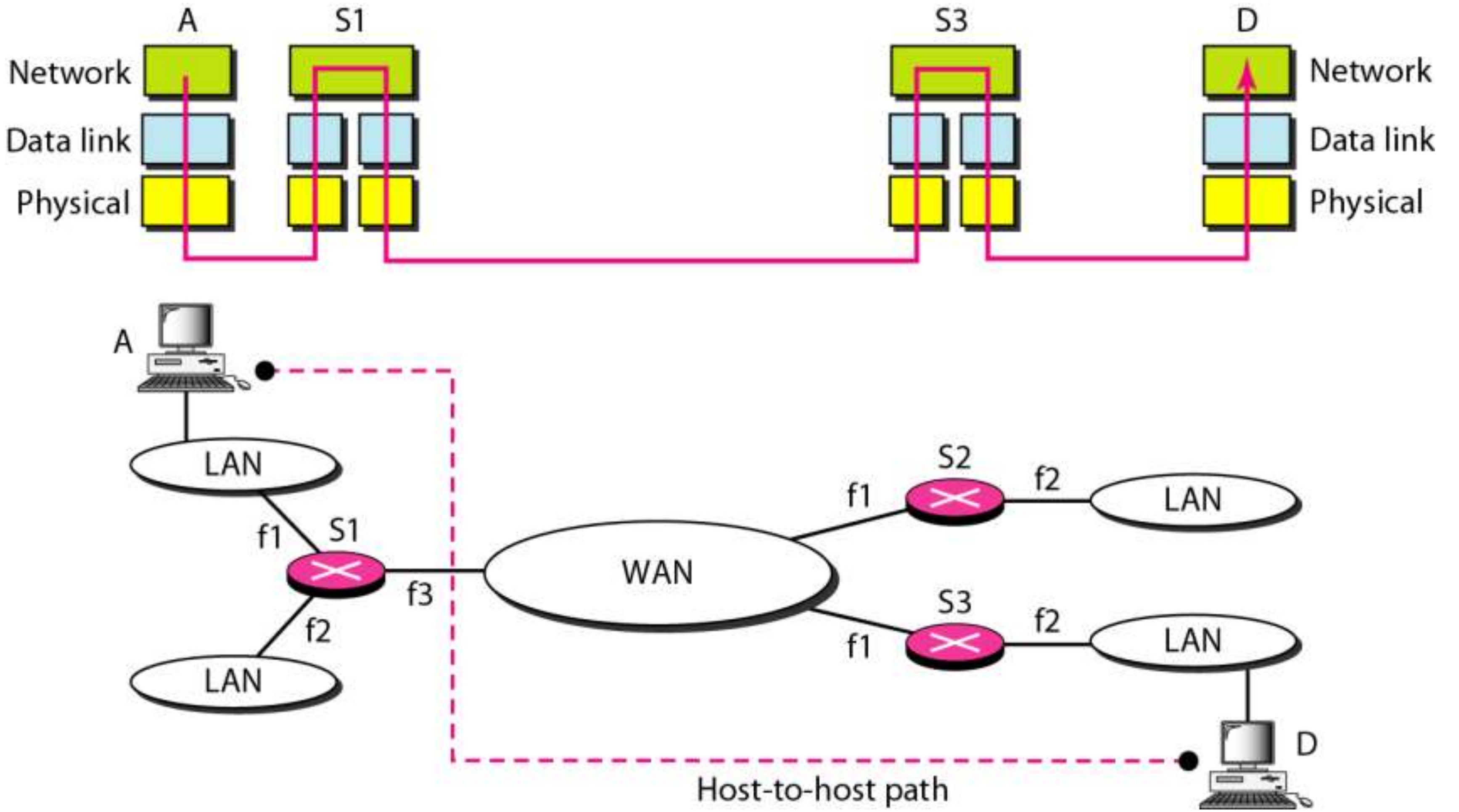


Figure 20.3 *Network layer at the source, router, and destination*

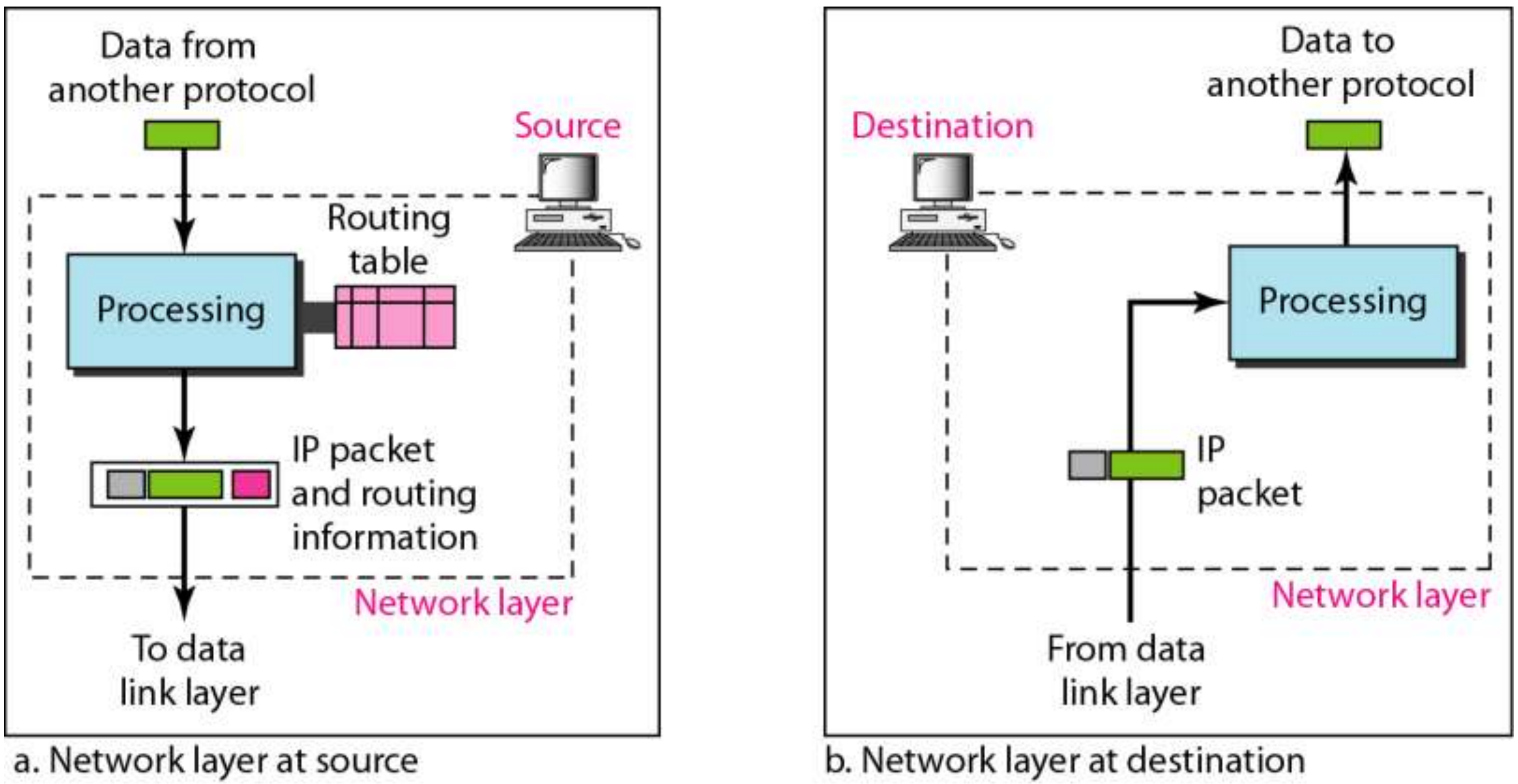
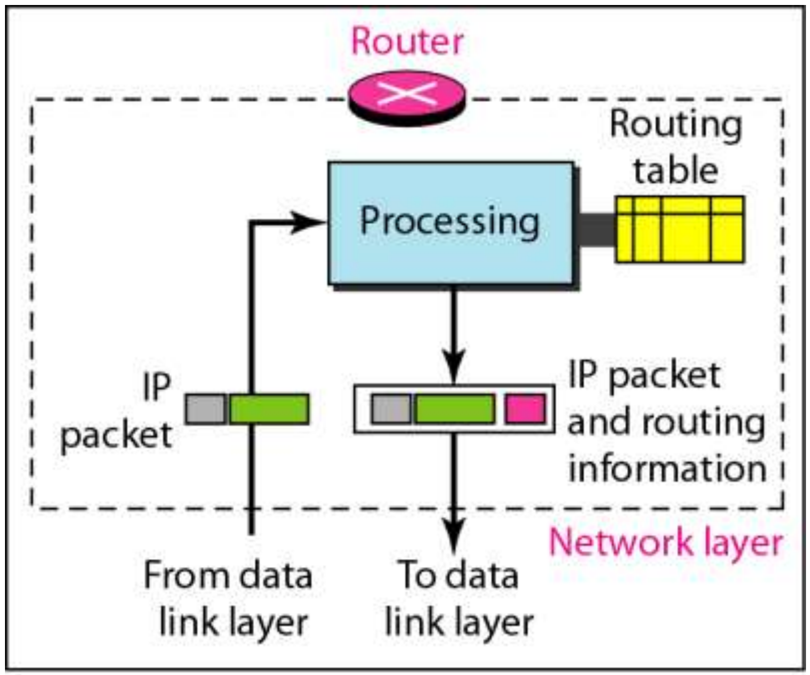


Figure 20.3 *Network layer at the source, router, and destination (continued)*



c. Network layer at a router

Note

Switching at the network layer in the Internet uses the datagram approach to packet switching.

Note

Communication at the network layer in the Internet is connectionless.

20-2 IPv4

The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols.

Topics discussed in this section:

Datagram
Fragmentation
Checksum
Options

Figure 20.4 *Position of IPv4 in TCP/IP protocol suite*

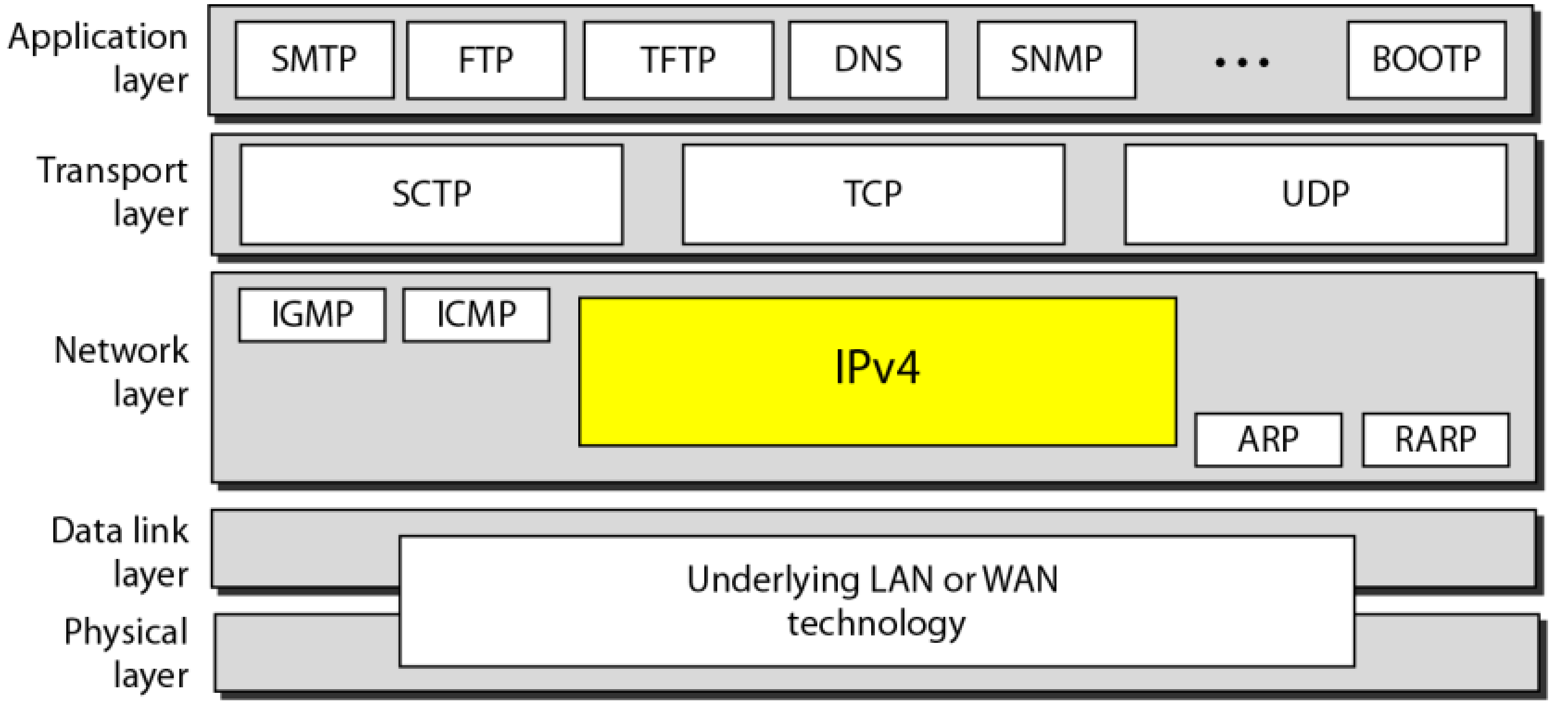


Figure 20.5 *IPv4 datagram format*

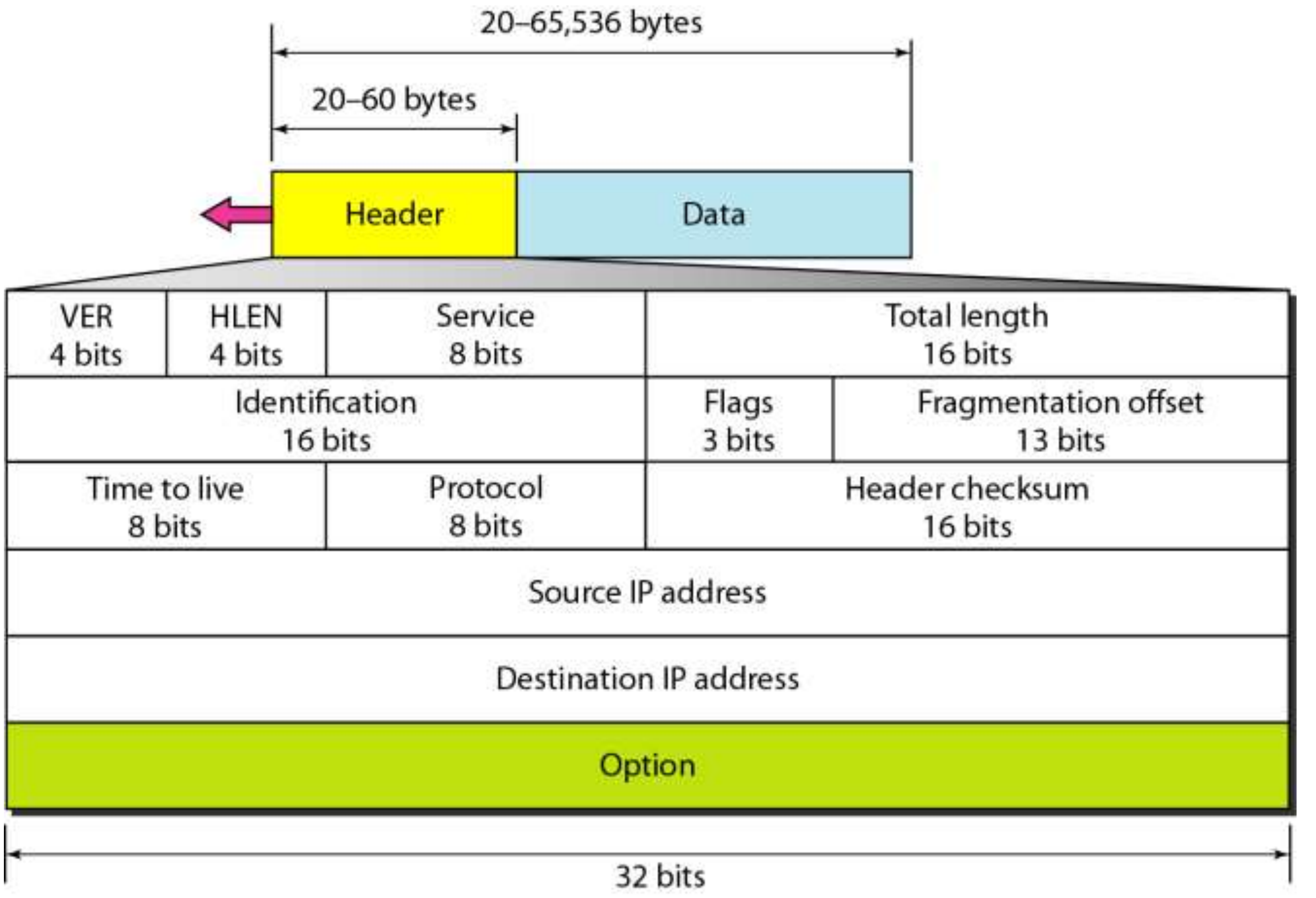
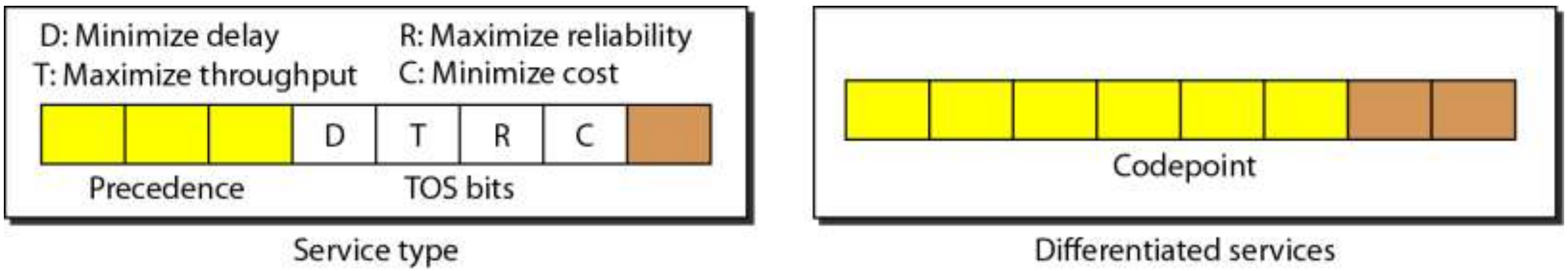


Figure 20.6 *Service type or differentiated services*



Note

The precedence subfield was part of version 4, but never used.

Table 20.2 *Default types of service*

Protocol	*TOS Bits*	*Description*
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

Table 20.3 ***Values for codepoints***

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Note

The total length field defines the total length of the datagram including the header.

Figure 20.7 *Encapsulation of a small datagram in an Ethernet frame*

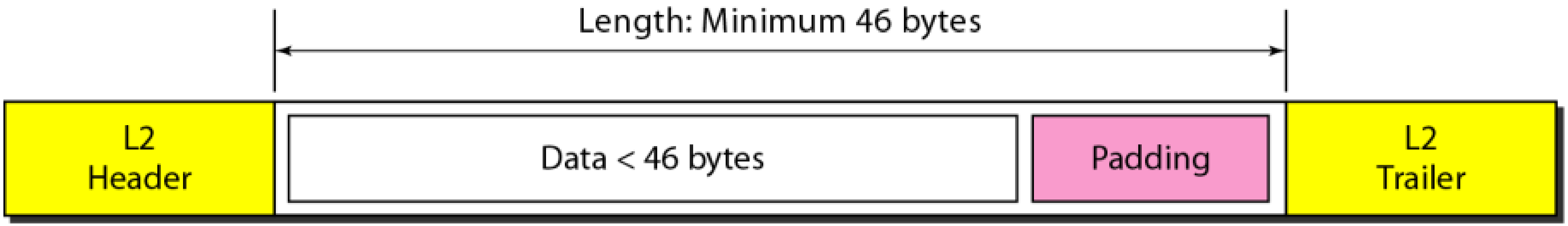


Figure 20.8 *Protocol field and encapsulated data*

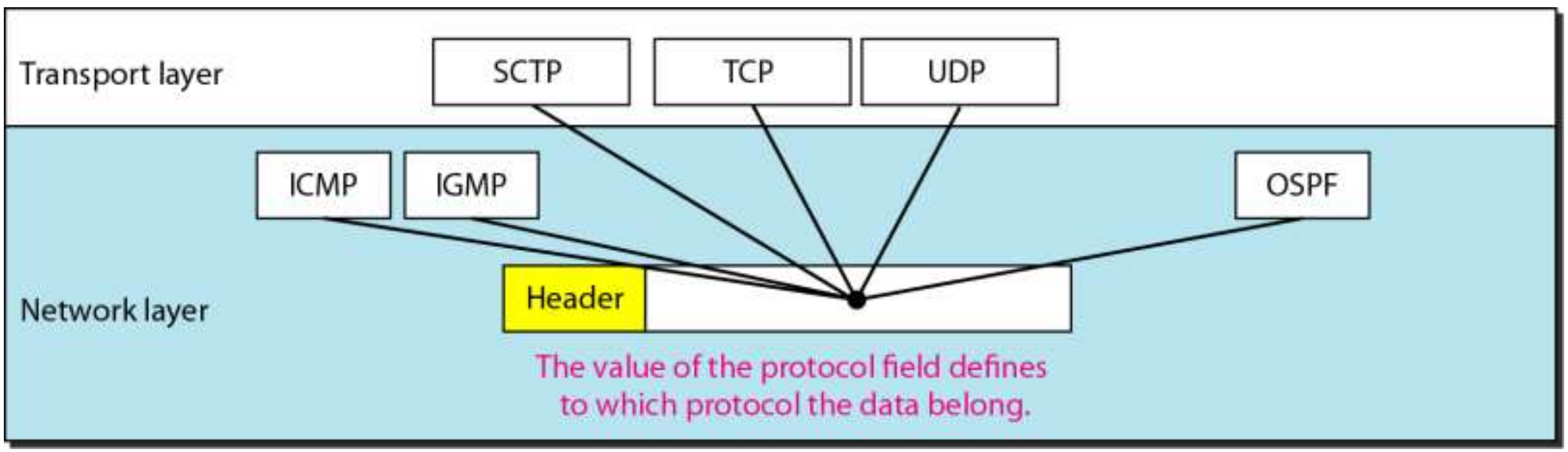


Table 20.4 ***Protocol values***

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Example 20.1

An IPv4 packet has arrived with the first 8 bits as shown:

01000010

The receiver discards the packet. Why?

Solution

There is an error in this packet. The 4 leftmost bits (0100) show the version, which is correct. The next 4 bits (0010) show an invalid header length (2 × 4 = 8). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Example 20.2

In an IPv4 packet, the value of HLEN is 1000 in binary. How many bytes of options are being carried by this packet?

Solution

The HLEN value is 8, which means the total number of bytes in the header is 8 × 4, or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.

Example 20.3

In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is 0x0028. How many bytes of data are being carried by this packet?

Solution

The HLEN value is 5, which means the total number of bytes in the header is 5 × 4, or 20 bytes (no options). The total length is 40 bytes, which means the packet is carrying 20 bytes of data (40 − 20).

Example 20.4

An IPv4 packet has arrived with the first few hexadecimal digits as shown.

0x45000028000100000102 . . .

How many hops can this packet travel before being dropped? The data belong to what upper-layer protocol?

Solution

To find the time-to-live field, we skip 8 bytes. The time-to-live field is the ninth byte, which is 01. This means the packet can travel only one hop. The protocol field is the next byte (02), which means that the upper-layer protocol is IGMP.

Figure 20.9 *Maximum transfer unit (MTU)*

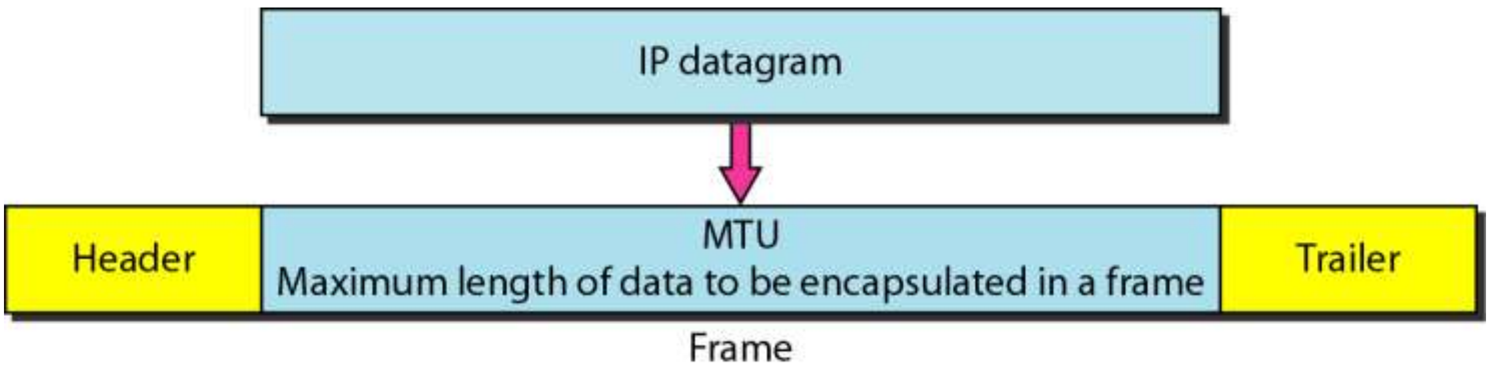


Table 20.5 ***MTUs for some networks***

Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

Figure 20.10 *Flags used in fragmentation*

Figure 20.11 *Fragmentation example*

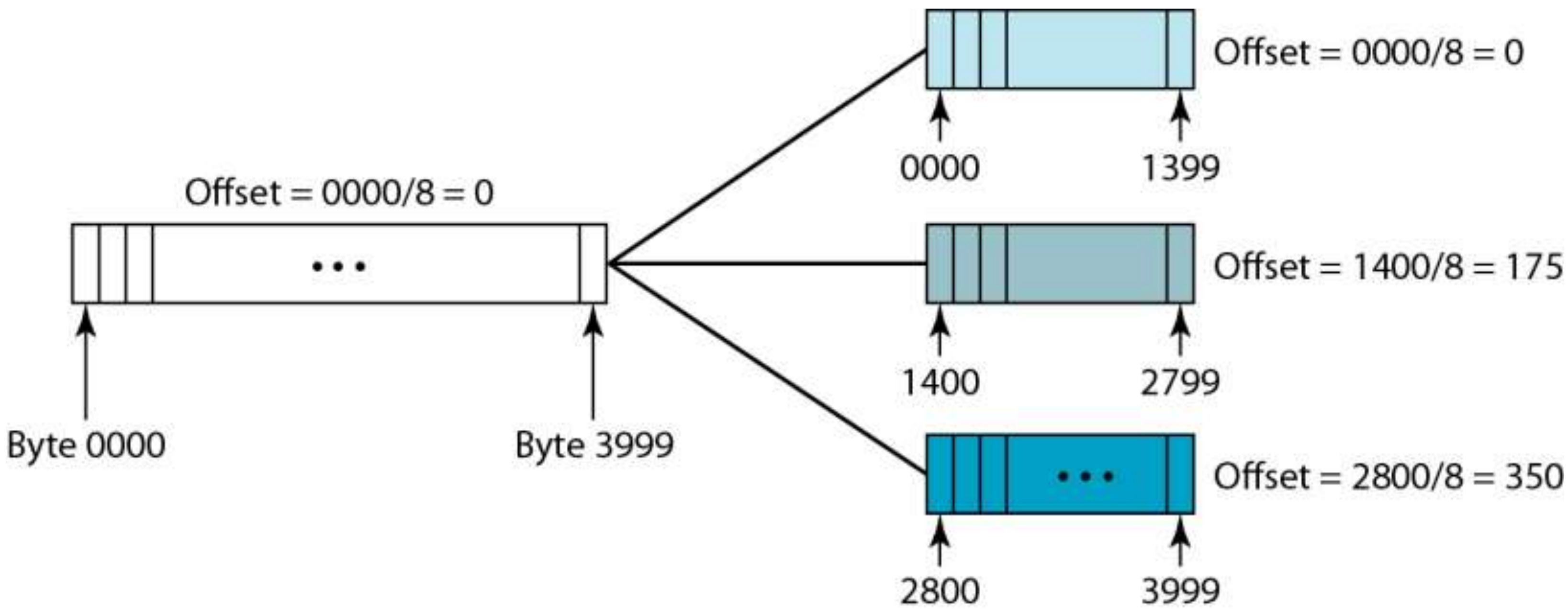
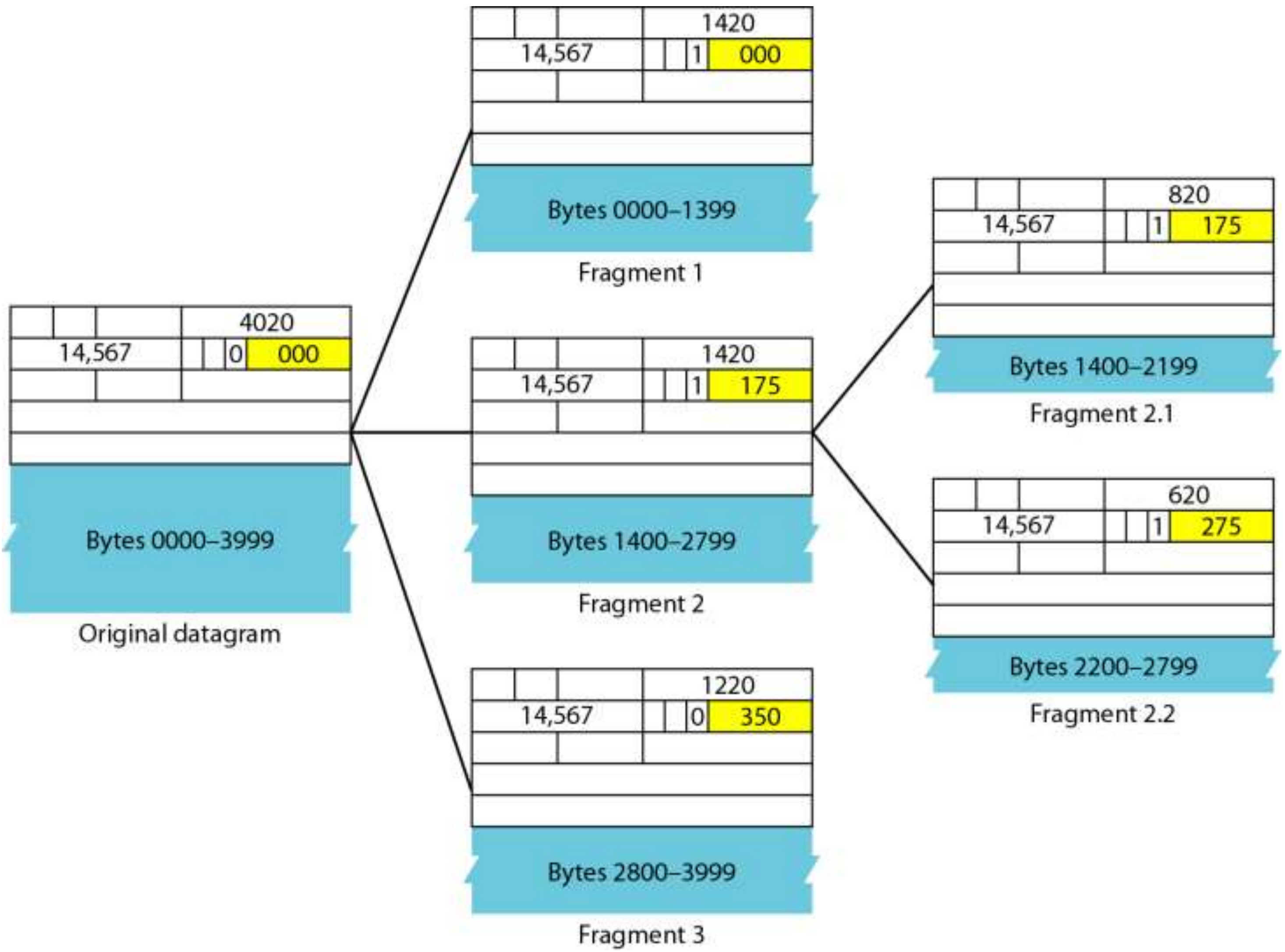


Figure 20.12 *Detailed fragmentation example*



Example 20.5

A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 0, it means that there are no more fragments; the fragment is the last one. However, we cannot say if the original packet was fragmented or not. A non-fragmented packet is considered the last fragment.

Example 20.6

A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 1, it means that there is at least one more fragment. This fragment can be the first one or a middle one, but not the last one. We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset).

Example 20.7

A packet has arrived with an M bit value of 1 and a fragmentation offset value of 0. Is this the first fragment, the last fragment, or a middle fragment?

Solution

Because the M bit is 1, it is either the first fragment or a middle one. Because the offset value is 0, it is the first fragment.

Example 20.8

A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?

Solution

To find the number of the first byte, we multiply the offset value by 8. This means that the first byte number is 800. We cannot determine the number of the last byte unless we know the length.

Example 20.9

A packet has arrived in which the offset value is 100, the value of HLEN is 5, and the value of the total length field is 100. What are the numbers of the first byte and the last byte?

Solution

The first byte number is $100 \times 8 = 800$. The total length is 100 bytes, and the header length is 20 bytes (5×4), which means that there are 80 bytes in this datagram. If the first byte number is 800, the last byte number must be 879.

Example 20.10

Figure 20.13 shows an example of a checksum calculation for an IPv4 header without options. The header is divided into 16-bit sections. All the sections are added and the sum is complemented. The result is inserted in the checksum field.

Figure 20.13 *Example of checksum calculation in IPv4*

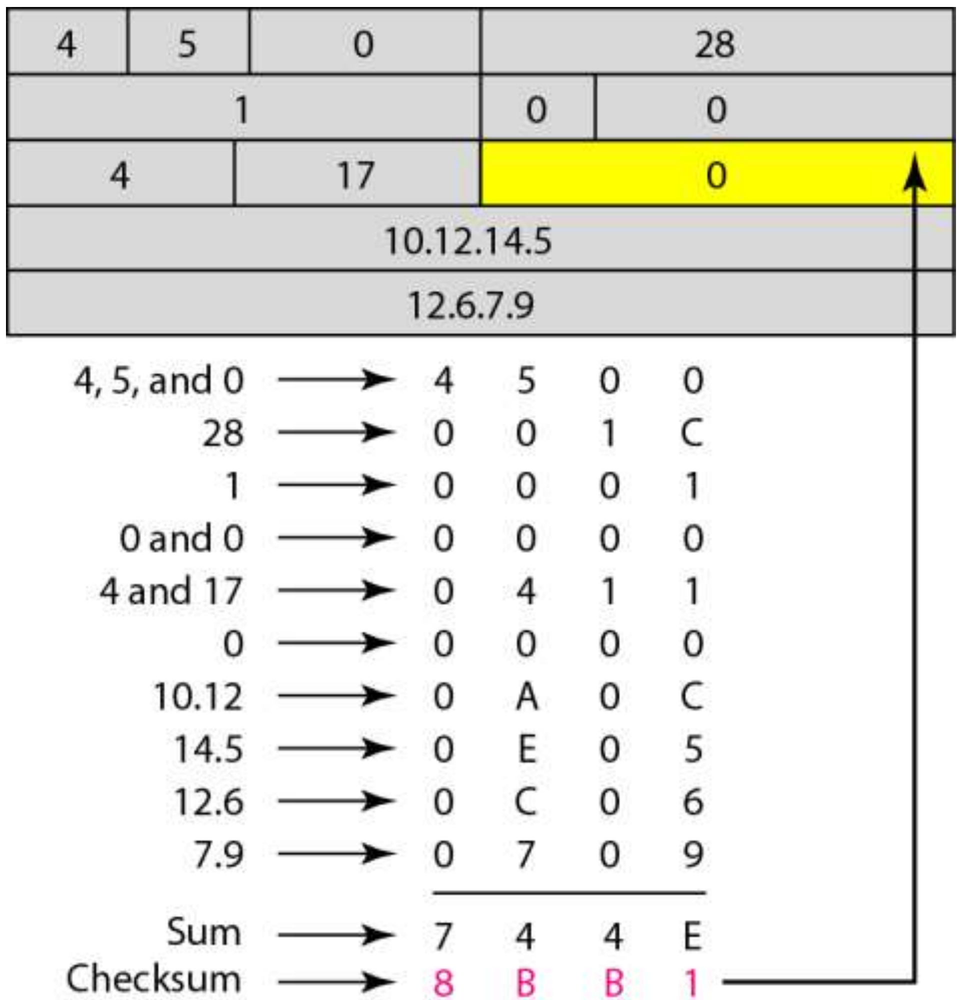
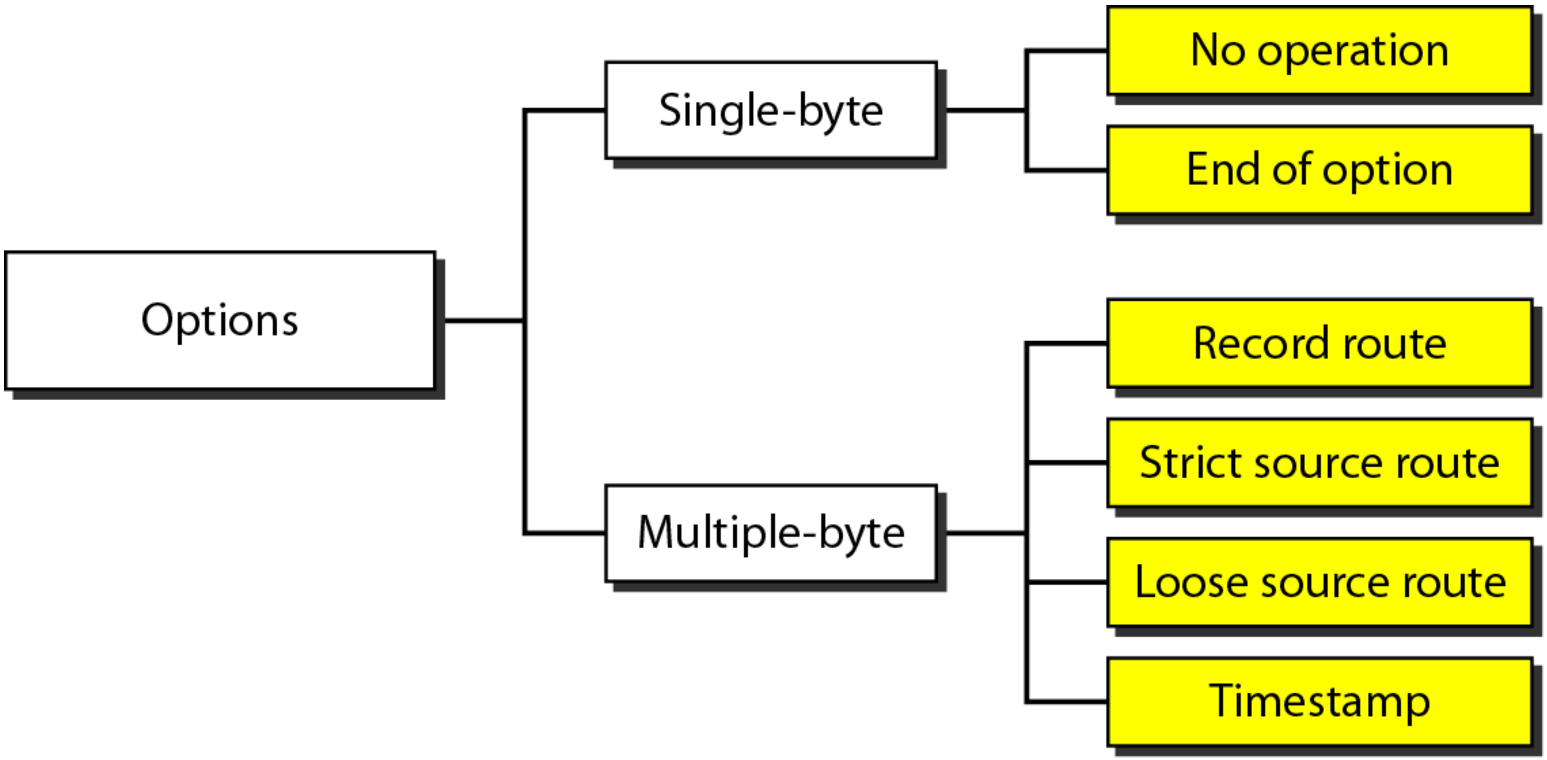


Figure 20.14 *Taxonomy of options in IPv4*



20-3 IPv6

The network layer protocol in the TCP/IP protocol suite is currently IPv4. Although IPv4 is well designed, data communication has evolved since the inception of IPv4 in the 1970s. IPv4 has some deficiencies that make it unsuitable for the fast-growing Internet.

Topics discussed in this section:

Advantages
Packet Format
Extension Headers

IPv6

Advantages

The next-generation IP, or IPv6, has some advantages over IPv4 that can be summarized as follows:

- Larger address space. An IPv6 address is 128 bits long. Compared with the 32-bit address of IPv4, this is a huge (296) increase in the address space.
- Better header format. IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the upper-layer data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
- New options. IPv6 has new options to allow for additional functionalities.
- Allowance for extension. IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- Support for resource allocation. In IPv6, the type-of-service field has been removed, but a mechanism (called low *label)* has been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.
- Support for more security. The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

Figure 20.15 *IPv6 datagram header and payload*

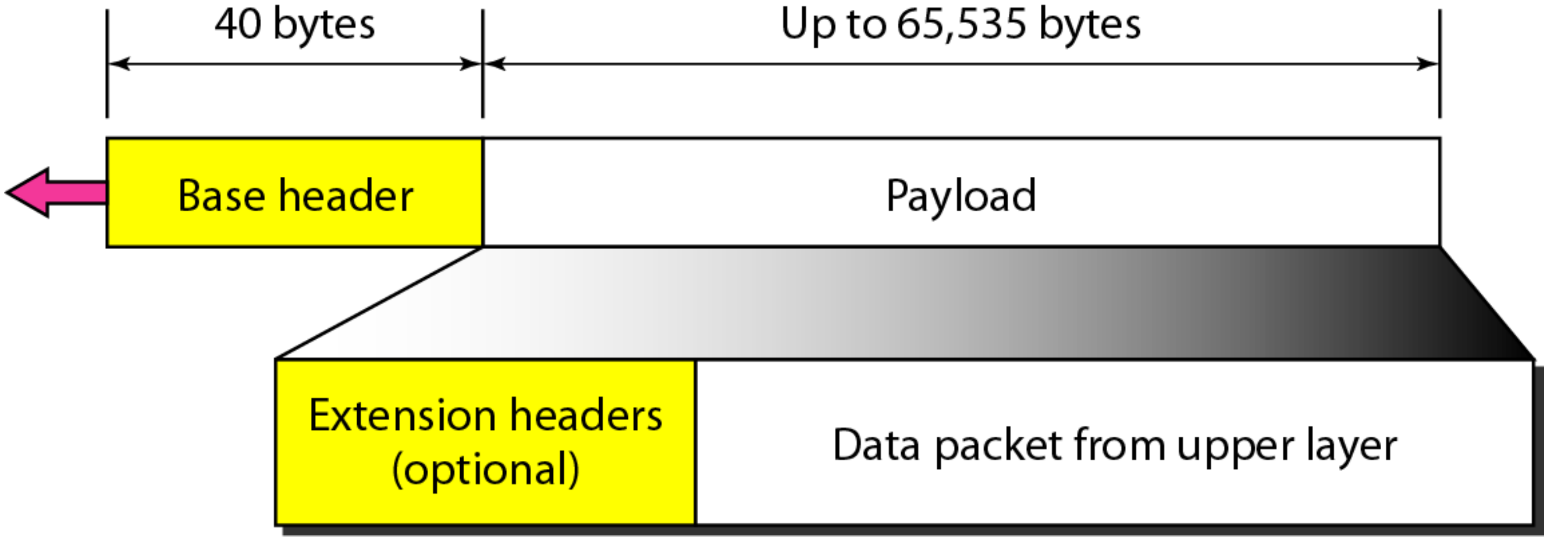
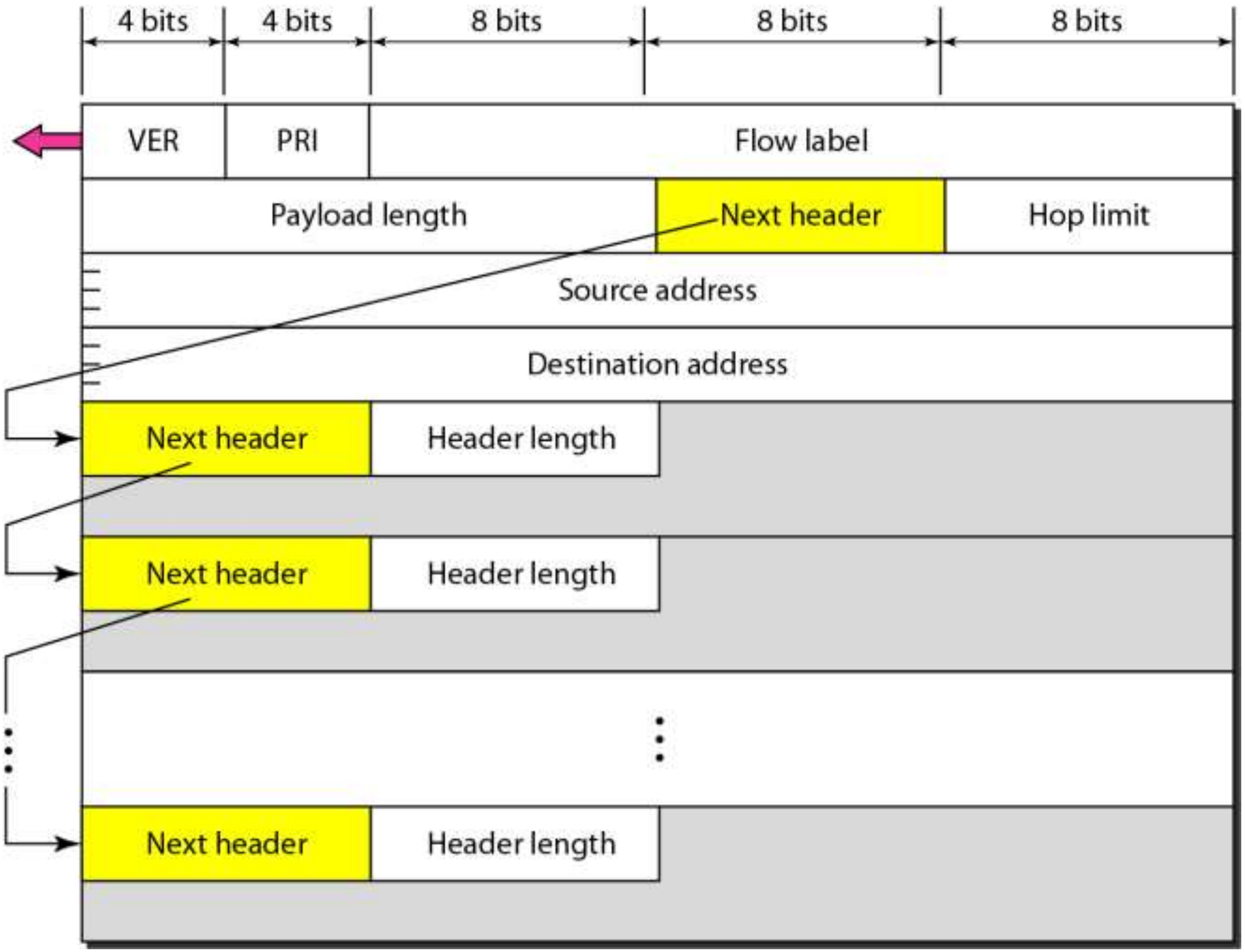


Figure 20.16 *Format of an IPv6 datagram*



Format of an IPv6 datagram

Base Header
These fields are as follows:
Version.
This 4-bit field defines the version number of the IP. For IPv6, the value is 6.
Priority.
The 4-bit priority field defines the priority of the packet with respect to traffic congestion. We will discuss this field later.
Flow label.
The flow label is a 3-byte (24-bit) field that is designed to provide special handling for a particular flow of data. We will discuss this field later.
Payload length.
The 2-byte payload length field defines the length of the IP datagram excluding the base header.
Next header.
The next header is an 8-bit field defining the header that follows the base header in the datagram. The next header is either one of the optional extension headers used by IP or the header of an encapsulated packet such as UDP or TCP. Each extension header also contains this field.
Hop limit.
This 8-bit hop limit field serves the same purpose as the TIL field in IPv4.

Table 20.6 *Next header codes for IPv6*

Code	*Next Header*
0	Hop-by-hop option
2	ICMP
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null (no next header)
60	Destination option

Table 20.7 ***Priorities for congestion-controlled traffic***

Priority	*Meaning*
0	No specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

Table 20.8 ***Priorities for noncongestion-controlled traffic***

Priority	*Meaning*
8	Data with greatest redundancy
. . .	. . .
15	Data with least redundancy

Table 20.9 *Comparison between IPv4 and IPv6 packet headers*

Comparison
1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
3. The total length field is eliminated in IPv6 and replaced by the payload length field.
4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
5. The TTL field is called hop limit in IPv6.
6. The protocol field is replaced by the next header field.
7. The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level.
8. The option fields in IPv4 are implemented as extension headers in IPv6.

Figure 20.17 *Extension header types*

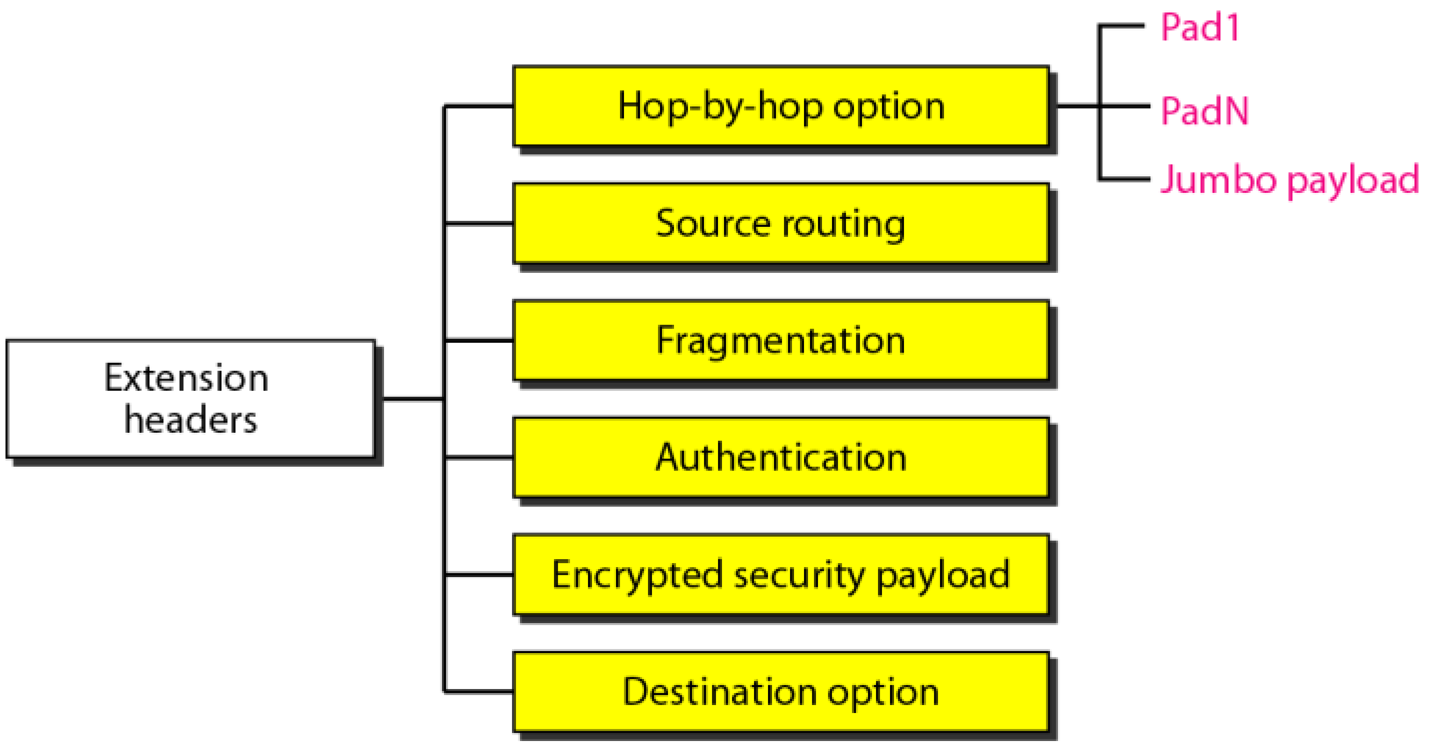


Table 20.10 *Comparison between IPv4 options and IPv6 extension headers*

Comparison
1. The no-operation and end-of-option options in IPv4 are replaced by Pad1 and PadN options in IPv6.
2. The record route option is not implemented in IPv6 because it was not used.
3. The timestamp option is not implemented because it was not used.
4. The source route option is called the source route extension header in IPv6.
5. The fragmentation fields in the base header section of IPv4 have moved to the fragmentation extension header in IPv6.
6. The authentication extension header is new in IPv6.
7. The encrypted security payload extension header is new in IPv6.

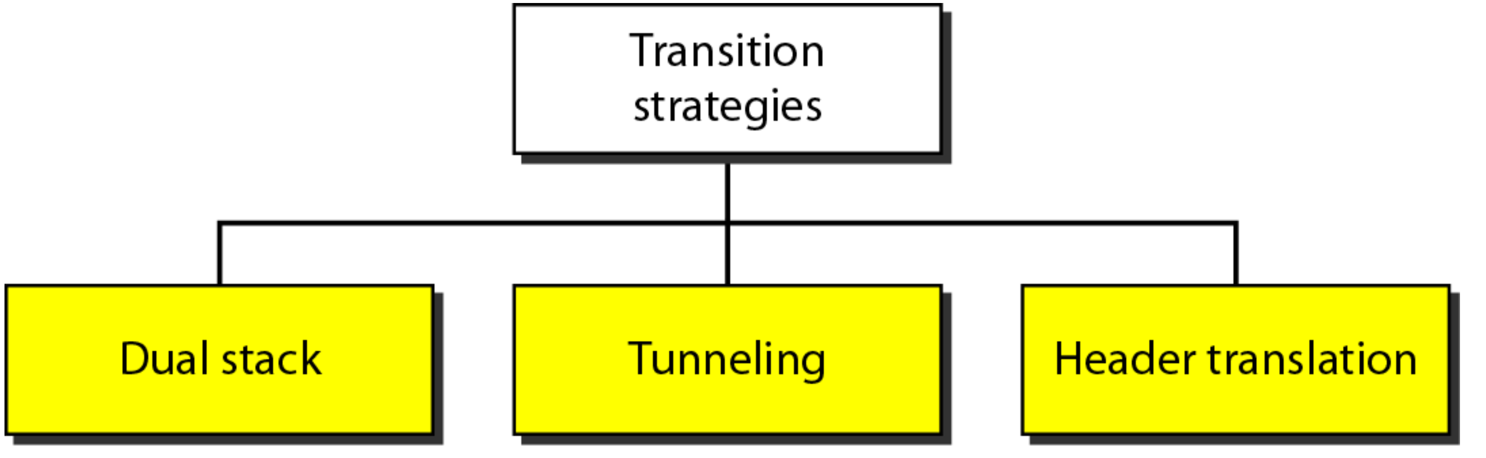
20-4 TRANSITION FROM IPv4 TO IPv6

Because of the huge number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly. It takes a considerable amount of time before every system in the Internet can move from IPv4 to IPv6. The transition must be smooth to prevent any problems between IPv4 and IPv6 systems.

Topics discussed in this section:

Dual Stack
Tunneling
Header Translation

Figure 20.18 *Three transition strategies*

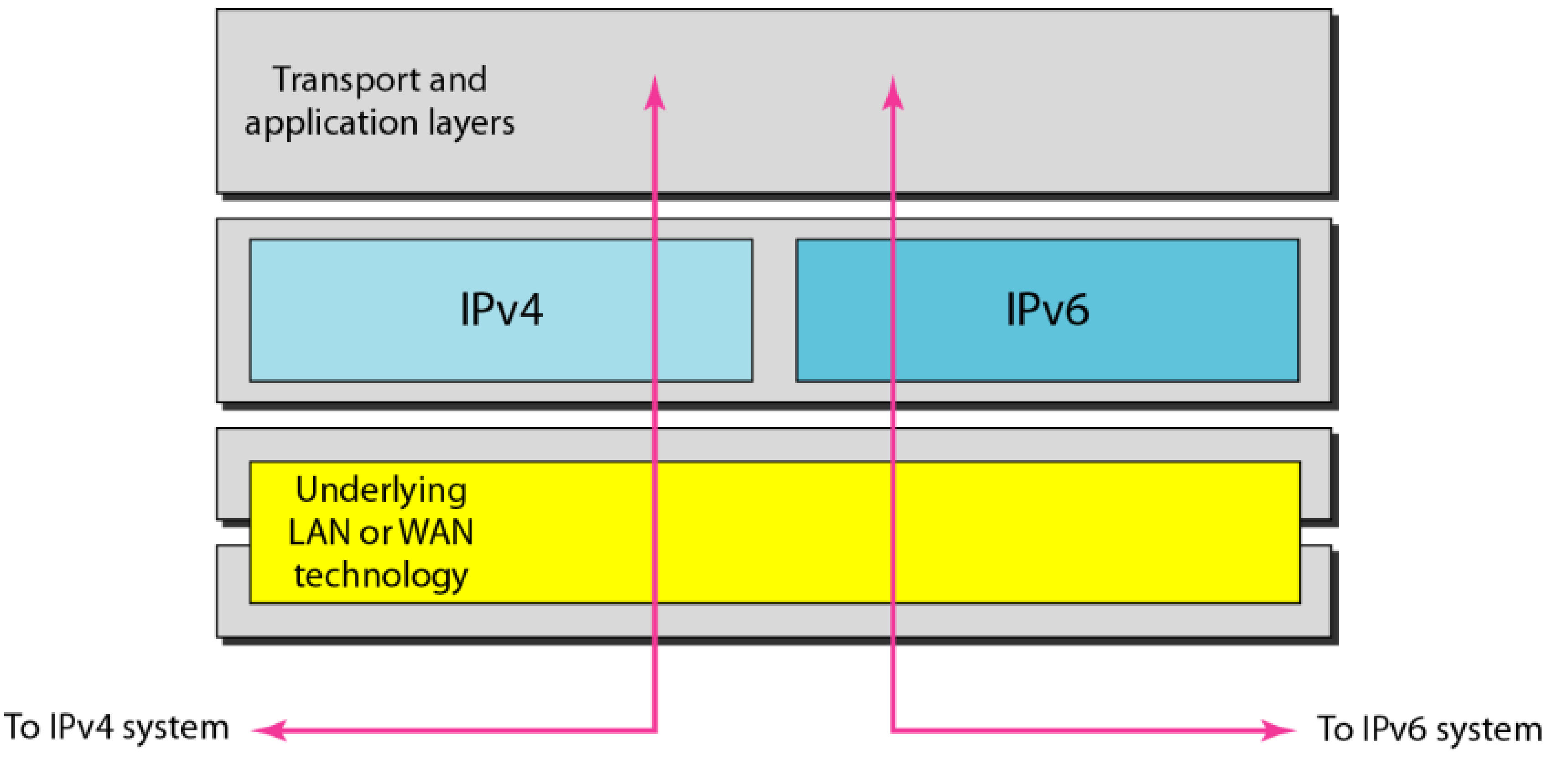


Dual stack

It is recommended that all hosts, before migrating completely to version 6, have a **dual** stack of protocols. In other words, a station must run IPv4 and IPv6 simultaneously until all the Internet uses IPv6.

To determine which version to use when sending a packet to a destination, the source host queries the DNS. If the DNS returns an IPv4 address, the source host sends an IPv4 packet. If the DNS returns an IPv6 address, the source host sends an IPv6 packet.

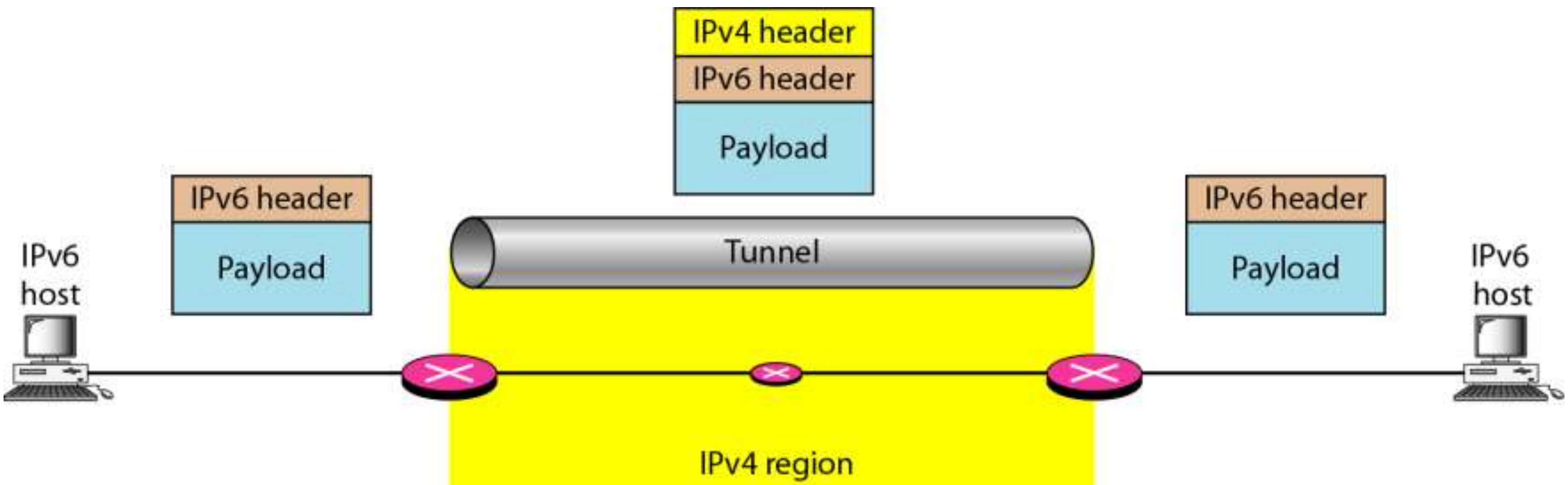
Figure 20.19 ***Dual stack***



Tunneling strategy

Tunneling is a strategy used when two computers using IPv6 want to communicate with each other and the packet must pass through a region that uses IPv4. To pass through this region, the packet must have an IPv4 address. So the IPv6 packet is encapsulated in an IPv4 packet when it enters the region, and it leaves its capsule when it exits the region. It seems as if the IPv6 packet goes through a tunnel at one end and emerges at the other end. To make it clear that the IPv4 packet is carrying an IPv6 packet as data, the protocol value is set to 41.

Figure 20.20 *Tunneling strategy*



Header translation strategy

Header translation is necessary when the majority of the Internet has moved to IPv6 but some systems still use IPv4. The sender wants to use IPv6, but the receiver does not understand IPv6. Tunneling does not work in this situation because the packet must be in the IPv4 format to be understood by the receiver. In this case, the header format must be totally changed through header translation. The header of the IPv6 packet is converted to an IPv4 header

Figure 20.21 *Header translation strategy*

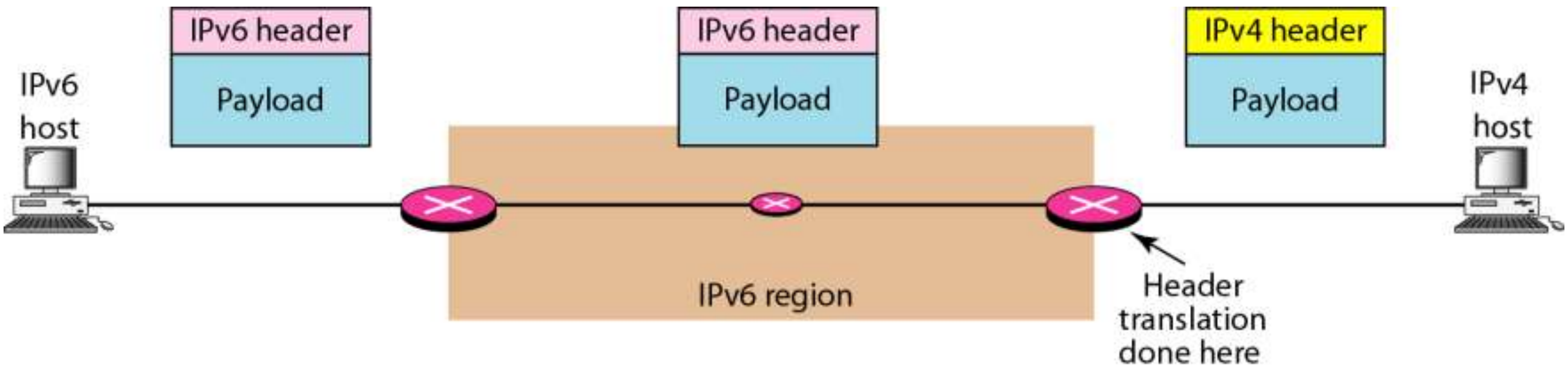


Table 20.11 *Header translation*

Header Translation Procedure
1. The IPv6 mapped address is changed to an IPv4 address by extracting the rightmost 32 bits.
2. The value of the IPv6 priority field is discarded.
3. The type of service field in IPv4 is set to zero.
4. The checksum for IPv4 is calculated and inserted in the corresponding field.
5. The IPv6 flow label is ignored.
6. Compatible extension headers are converted to options and inserted in the IPv4 header. Some may have to be dropped.
7. The length of IPv4 header is calculated and inserted into the corresponding field.
8. The total length of the IPv4 packet is calculated and inserted in the corresponding field.